OS: Part 1

OS: 1) Program which manages all other programs in a computer

2) interface between hardware and application software

user view

1) varies by the interface being used.

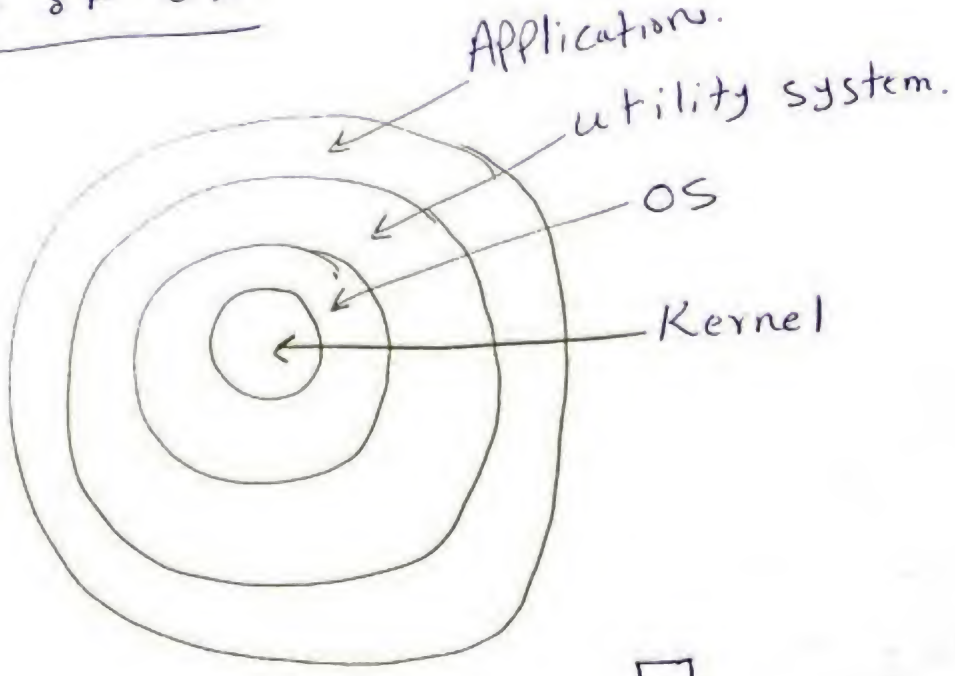
2) sys. Consists of monitor, Keyboard and mouse

system views

1) we can view an OS ass resource allocator.

2) OS is program that is most intimate with hardware.

structure of OS



* Kernel

1) the care of os. 2) called real time executive.
3) Contains all the devices that interact with the hardware.

*Os

→ Controls and coordinates the use of hardware among the various applications programs for various users.

*utility system

→ system utilies are program that perform individual, specialized management tasks.

*Applications

→ such as word processor, spreadsheet.
→ compilers define the way in which these resources are used

Types of operating system

1) single user

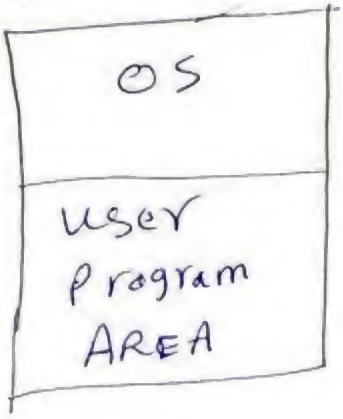
→ only one user can operate on single "os" such as "Ms-Dos"

2) Multi user

→ users can operate of single operating system.
→ ~~also~~ allow many different users to take advantage of Computer resources. for example "windows"

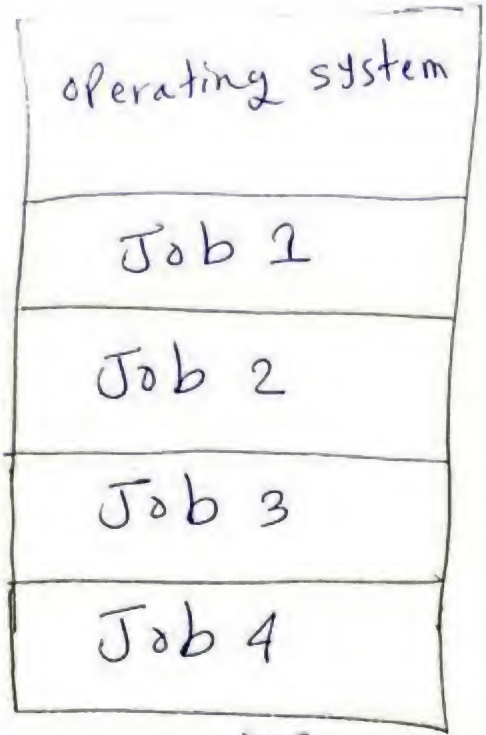
3) Batch system

→user Can execute Programs in batch form.
→each batch Contains similar type of programs.
→batch is made by server & executed by OS.



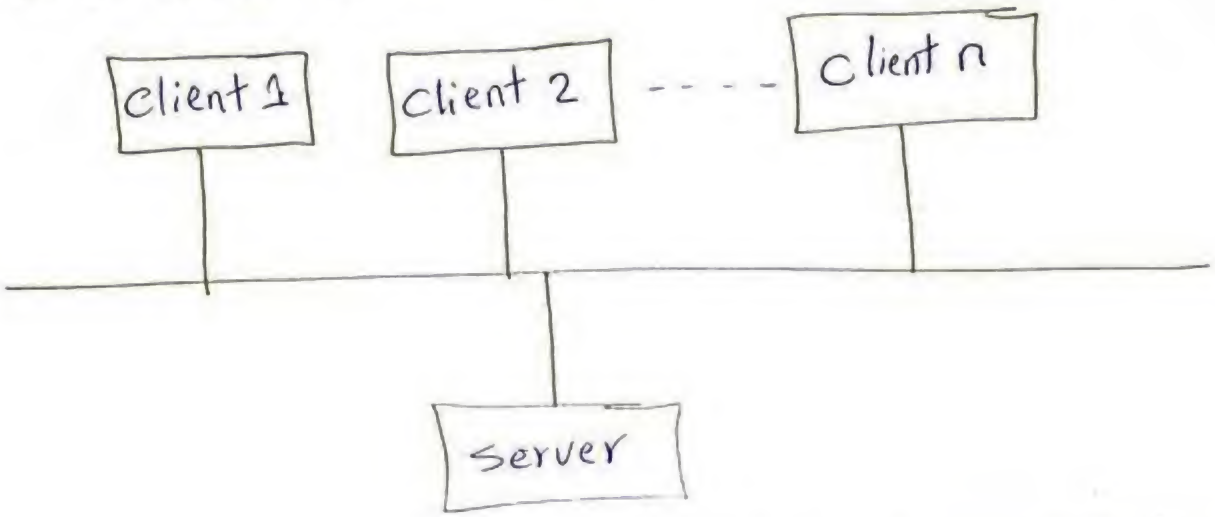
4 MultiProgramming system

→OS Keeps several Programs in memory at the same time.
→increase CPU utilization by organizing Job so that CPU always has one to executed



5) Peer to Pear Systems

→ Systems are connected with straight line and first system will be consider as server and other systems are knows as a clients.



6) Distributed system

→ It is similar to client server architecuture.

→ clients request the server for their requirements and server would provide all servers that is needed to clients.

→ allow multiple app. program to cooperate to solve participate task.

7) Real time :-

→ Consists the time which is requested to the execution of programs.

→ two types (hard time - soft time)

Services to operating system

* Program execution

→ system must be able to load a program into memory and to run that program.
→ Program must be able to end it's execution. normally or abnormally.

* I/o operations

→ running program may be required Ilo.
→ os must provide a means to do.

* File system manipulation

→ the file system is of need to read and write file.
→ Program also need to create and delete file by name.

* Communications

→ It may be implemented via shared memory or by the technique of message passing.

* Error detection

→ os needs to be aware of possible error.
→ Error may be occur in CPU and memory hardware.

* Accounting

→ the record of OS Keeping may be used for accounting or simply for accumulating uses statics.

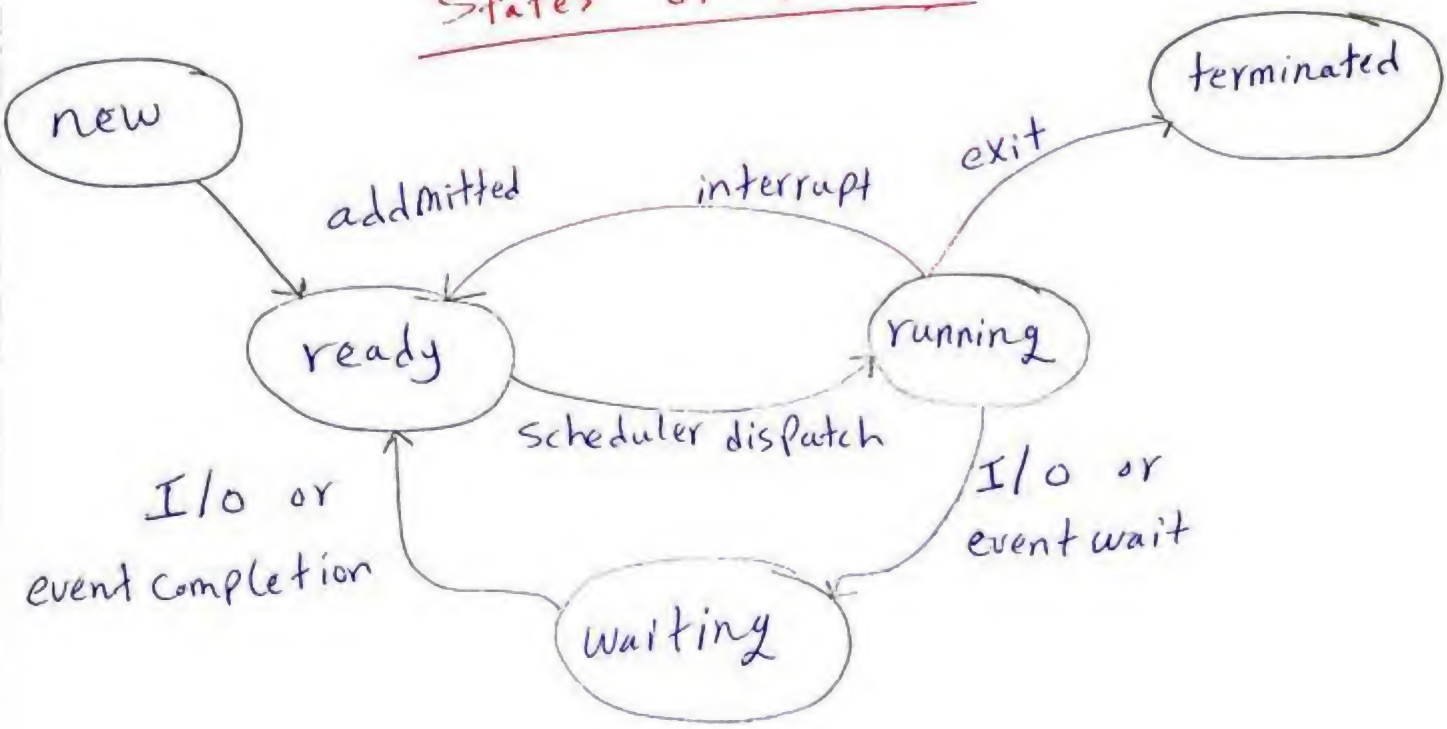
Process

→ this word was first used by designers of Multics in 1960.

→ used some what interchange with task or Job.

→ Process means when a program in execution.

States of process



New → Process is being created.

Running → instructions are being executed.

Waiting → Process is waiting for some event to occur.

Ready → Process is waiting to be assigned to a Processor.

Terminated → Process has finished executions.

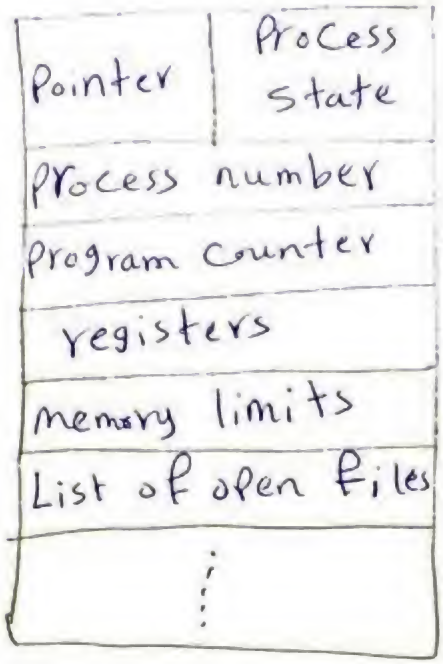
⇒ As a process executed, it changes state. the state of a process is defined in part by the current activity of process

Process Control Block

→ Process are represented by PCB.

→ Schedular handles Processes maintained in form of PCB.

→ It acts as Complete record to process & has the following parts.



« Process Control Block"

→ Process state

↳ state may be new, ready, running, waiting and so on.

* Program Counter

→ PCB Contains the address of next instruction to be executed.

* CPU Registers

→ registers vary in number and type, depending on Computer architecture.

→ They include accumulators, index register and stack pointer.

* Memory-management Information

→ PCB holds value of base & limit register, Page table or depending on the memory system used by OS.

* I/o status information

→ information includes list of i/o devices allocated to this process a list of open files etc allocated to the process is stored in PCB.

Scheduling

→ fundemental operating system function.

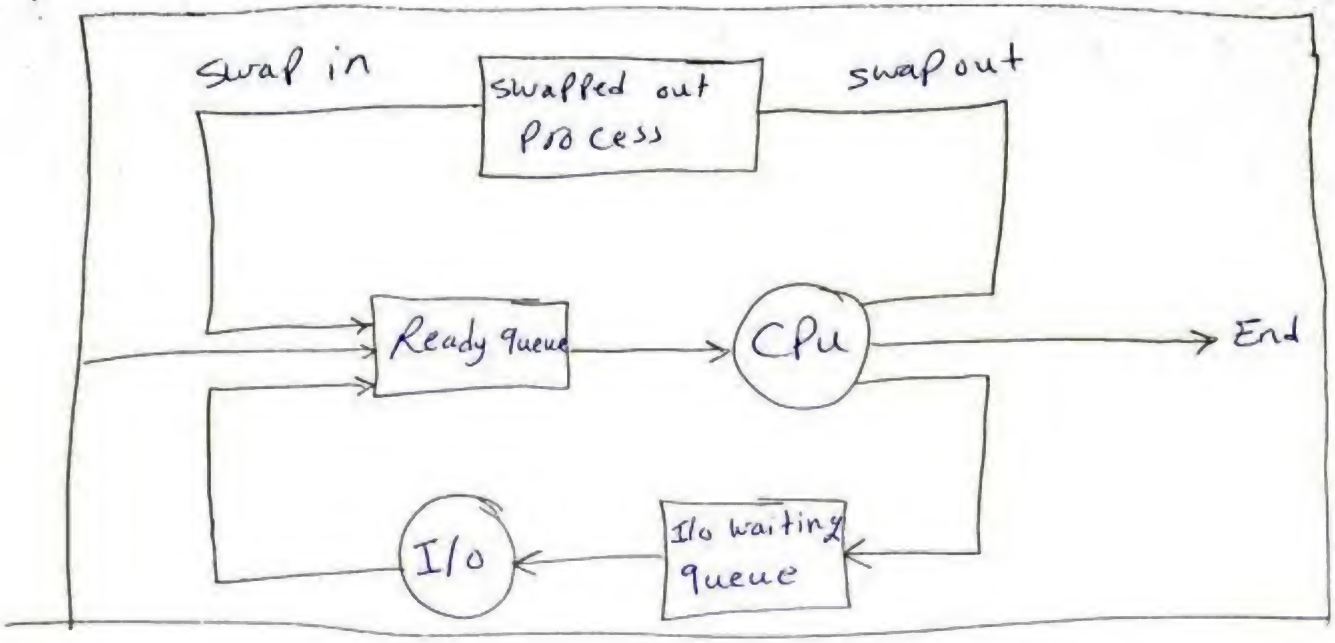
→ every Computer resource is scheduled before use.

→ is classified as

1) long term :- which Process to be admit.

2) medium term :- which process to swap in or out.

3) short term :- which read Process to execute next.



Inter Process - Communication (IPC)

→ means of information sharing and synchronization between process

→ In Computing, IPC is set of methods for the exchange of data among multiple threads in one or more process.

→ IPC in multiprocessor system. This allows a program to handle many user requests at the same time.

Threads

→ Processes that share an address space.

→ each thread has a program Counter, register and run time stack.

→ There are multiple threads of control and single address space.

→ A threads sometimes called a lightweight process (LWP) is a basic unit of CPU utilization.

→ traditional or heavy weight process has a single thread of control.

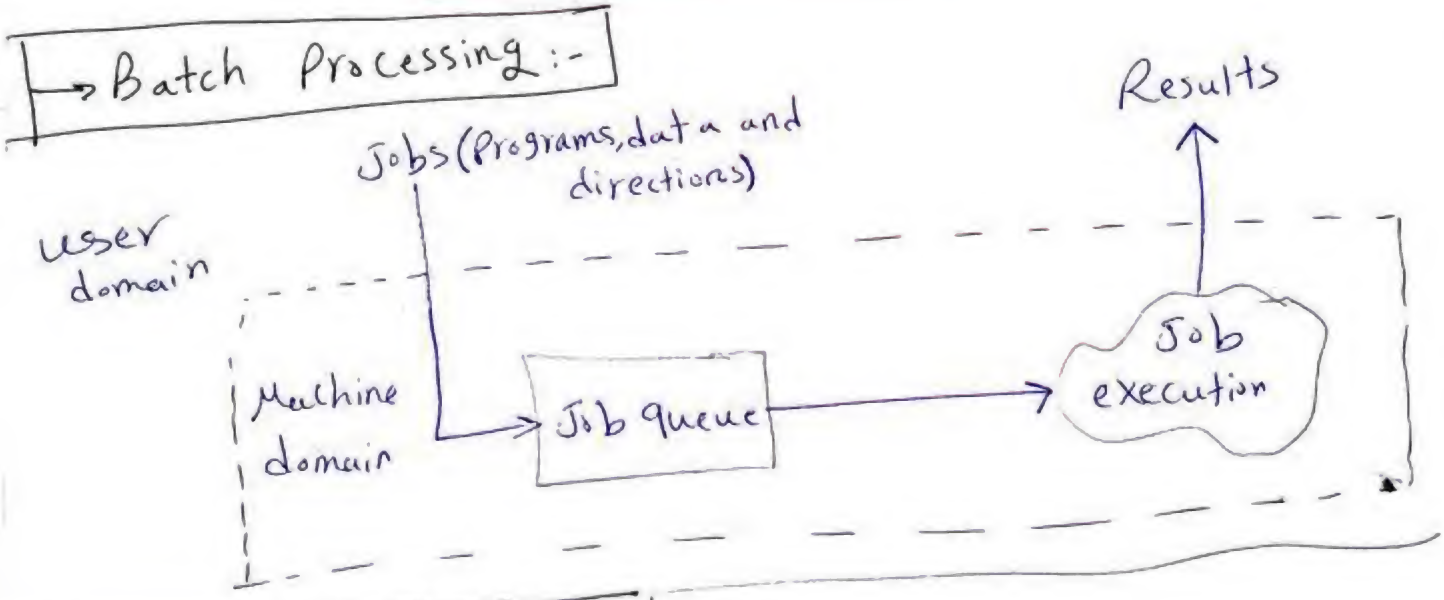
OS : Part 2

→ Functions of OS

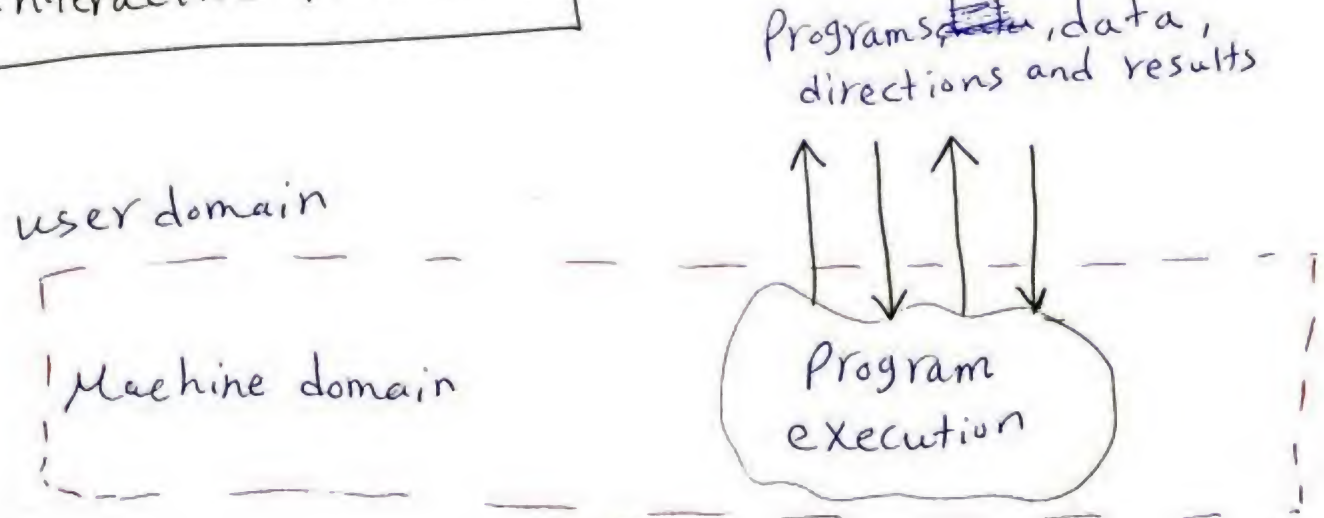
1) oversee operation of Compiler.
2) store and retrieve files.
3) schedule Programs for execution.
4) Coordinate the execution of Programs.

Shared Computation

→ Batch Processing :-



Interactive Processing



Time sharing / MultiTasking

1) users are seeking services from same machine at the same time (time sharing)

2) Multiple terminals connected to same machine.

3) When multiprogramming is applied to single-user environments is called multitasking.

Multi Processor operating systems

1) Provide time sharing / multi-tasking capabilities by assigning different tasks to different processors as well as sharing the time on single processor.

2) Problems to solve

→ Load balancing: dynamically allocating tasks to the various processor so that all of them are used efficiently.

→ Scaling: breaking tasks into sub-tasks compatible with number of processors available.

3) Trend to develop a network wide "OS" rather than networks of individual "OS".

Embedded operating systems

1) used in hand held devices, mobile phones, Cars, etc...

2) limited data storage and Power Conservation are the big challenges.

3) Examples VxworKs, PalmOS, RamDos, Symbinan.

Types of software

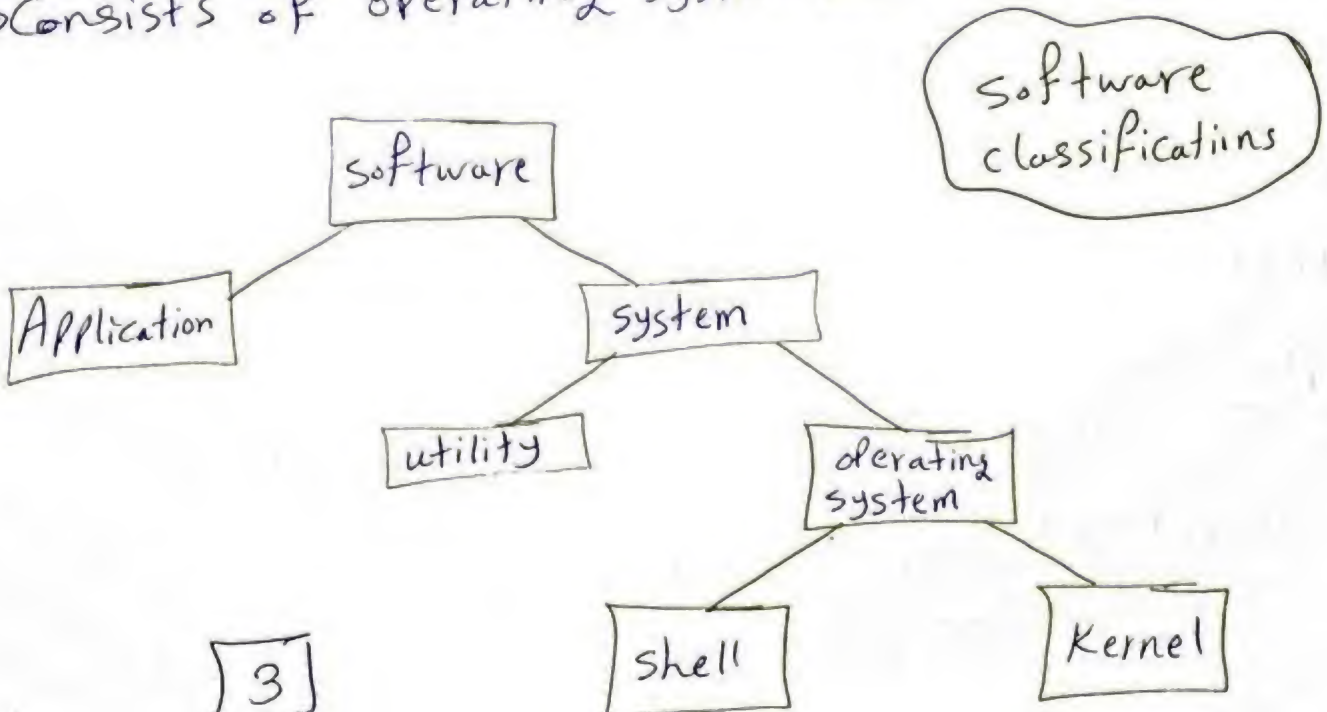
1) Application software

→perform specific tasKs for users: spread sheets, database systems, Program development, games, ... etc

2) system software

→Provide infrastructure for application software

→Consists of operating system and utility software



OS Components

*shell

→ communicates with users. → text based

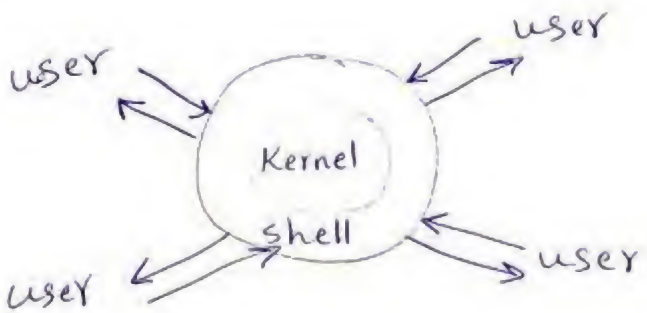
→ Graphical user interface (GUI)

*Kernel : Performs basic functions

→ file manager → device drivers

→ memory manager. → process manager (scheduler, etc)

⇒ shell is an interface between users and "OS".



*file manager

*Role ⇒ Coordinate the use of machine's mass storage facilities.

*Hierarchical organization

→ Directory: user-created bundle of files and other directories

→ Directory Path: sequence of directories within directories

*Access/operations to files is provided by file ~~manager~~ manager via a file descriptor

Device manager

1) Part of OS Presented as collection of device drivers.
2) Specialized software that communicate with the controllers to carry out operations on ~~the~~ peripheral devices connected to computer.
3) each driver is specifically designed for its type of device.

Memory manager

1) Coordinating the use of main memory (allocate and deallocate space in main memory)
2) when total required memory space exceeds the physical available space.

~> may create illusion that machine has more memory than it actually does (virtual memory)

Process manager

* scheduler

→ Part of Kernel in charge with the strategy for allocation / de-allocation of CPU to each completing process

→ Add new processes to the process table and removes completed processes from process table.

* Process table contains

1) memory area assigned to the process.
2) Priority of the process. 3) state of process (ready or wait)

* Dispatcher

→ component of Kernel that overseas execution of the scheduled processes.

→ Control allocation of CPU to the process in process table.

→ It performs "process switch" procedure to change from one process to another.

Process A → Dispatcher → Process B

Handling Competition for Resources

→ important task of OS is to allocate resources to the processes.

→ Semaphore → A control flag.

→ Mutual exclusion ⇒ Requirement for proper implementation of a critical region so that only one process at a time will execute sequence of instructions part of critical region

Deadlock

→ Another problem of resource Allocation:-

→ Processes block each other from continuing.

→ conditions required for deadlock.

a) Competition for non-sharable resources.

b) Resources requested on partial basis.

c) An Allocated resource can't be forcibly retrieved.

Security

→ one of the role of OS is to Provide security

→ Attacks from outside
  * Insecure passwords
  * Counter measures.
    → Auditing software.

→ Attacks from within.

Counter measures → Control Process activities via Privileged (modes and instructions).

---

* Booting

→ transfers OS from mass storage into the main memory.

* Bootstrap

→ Program in ROM.

→ Transfers OS from mass storage to main memory.

→ Executes Jump to "OS".

→ Run by CPU when Power is turned on.

Lec 3 : I/O devices

*input device :-
transforms information from the user into data that Computer application Can Process.

→ choice and method of use of Input device should Contribute Positively to usability of the system.

→ usability depends on the Provision of appropriate feed back.

⇒ Pointing devices (cursor control)

a) 3D tracker
- → Replaces mouse where desk space is limited (laptops)
- → Relays Position and orientation to a receiver.

b) Joystick
→ small stick, movable in any direction in a fixed socket.

c) Mouse
→ Relative device. Buttons for discrete input.

d) Trackball
→ Rotatable ball in fixed socket.

e) Tablet
→ used with Puck. Absolute device.

f) Mole → (foot mouse)

g) Dataglove

↳ Communicates hand and finger position to an app. used for manipulating virtual objects.

h) Touch-sensitive screen

↳ special screen that detects the position of a finger touching it.

e) Light Pen

↳ location found by beam passing through screen during refresh cycle.

Device	Advantages	Disadvantages
Touch-screen	→ no moving parts. → durable low price.	→ tiring for polonged use.
Light Pen	→ useful with flat screen → no training needed	→ not attractive by dark areas.
Mouse	fast, inexpensive, accurate.	→ requires flat, dedicated surface area.
Trackball	inexpensive, fine control, little disk space needed.	→ less easy to use than mouse.
Joystick	inexpensive, little desk space needed.	→ Can be inconvenient to use if built in

*Choosing appropriate input devices:-

a) Matching devices with work.

→ natural mapping between use, feedback, meaning of result and user's mental model are needed.

→ Particular manipulations needed to accomplish a piece of work need to be analyzed.

b) Matching devices with users.

c) Matching devices with environment of use.

→ space. → relation to other concurrent tasks etc.,

Developments in input

⇒ speech recognition

*advantages: minimal user training, freedoms of hands opportunities for physically disabled.

*disadvantages: recognition system often needs training, liable to error; difficult for app. to interpret human speech.

⇒ Handwritten input

*advantages:: easier to separate words, may need some training for users.

*disadvantages
↳ Cursive script difficult to read, system may need training.

output devices

→ Provides information or feedback in a form which is understandable by humains.

*

Visualisation

~~→ Dynamic visualisation~~

→ Dynamic Visualisation : became very important in information-rich applications.

→ The key issue is to find visual forms that support the users' mental model.

→ Perceptualisation → the multimedia equivalent of Visualisation.

→ 3D animation and virtual Reality are likely to increase in importance for "perceptual" interfaces.

Sound

→ It can complement a visual interface when users' attention is likely to turn away from a VDU screen.

→ important use → to deliver information on background events that need continual monitoring.

* "seven plus or minus two" rule for information overload applies to sounds, too.
→ sound can be important in interfaces for the visually disabled.

Digital speech techniques

* Concatenation:- involves:
a) digitally recording human speech in large chunks (word, short sentences)
b) reassembling it and play back.

e.g ⇒ talking clock.

*Synthesis-by-rule involves synthesizing speech according to prescribes rules of sound formation to generate more "natural-sounding" tone.

Multimedia

→ Potential for multimedia includes the fact that it is estimated that only 7% of business critical information is "record-based".
→ multimedia databases including photos, video clips sounds, etc., may permit digitalisation of much of the other 93%.

I/o Hardware

→ Incredible variety of I/o devices.

→ Common Concepts (Port - Bus - Controller)

→ I/o instructions control devices.

→ Devices have addresses used by

a) Direct I/o instructions.

b) Memory mapped I/o.

Polling

→ Determine state of device

(Command ready - busy - Error)

→ Busy - wait cycle ~~for~~ to wait for I/o from device.

Interrupts

→ interrupt handler receives interrupts.

→ interrupt mechanism also used for ~~exce~~ exceptions.

~~→ Maskable to ignore or delay some~~

→ maskable to ignore or delay some interrupts.

→ interrupt vector to dispatch interrupt to correct handler.

interrupt - Driven I/o Cycle ans ←

Page 7 Lec 3 Part 2 pdf

Direct memory Access

→ used to avoid Programmed I/o for large data movement.

→ Requires DMA Controller.

→ Bypasses CPU to transfer data directly between I/o device and memory.

→ six steps Processes to Perform DMA transfer. (on pdf)

~~Application I/o interface~~

Network devices

→ Varying enough from block and character to have own interface.

→ Approaches vary widely (Pipes, FIFo, Streams, mailboxes)

Clocks and Timers

→ Provide current time, elapsed time, timer.

→ if Programmable interval time used for timing, periodic interrupts.

→ ioctl (on UNIX) covers odd aspects of I/o such as clocks and timers.

Blocking and nonblocking I/O

*Blocking : Process suspended until I/O completed.

→ Easy to use and understand.
→ Insufficient for some needs.

*Non blocking

→ I/O Call returns as much as available.
→ Implemented via multithreading.
→ returns quickly with count of bytes ~~read~~ read or written.

*Asynchronous

→ Process runs while I/O executes.
→ Difficult to use.
→ I/O subsystem signals Process when I/O Completed.

Kernel I/O Subsystem

*scheduling

→ Some OSs try fairness.
→ Some I/O request ordring via per-device queue.

*Buffering store data in memory while transfer between devices.

a) to maintain copy semantics.
b) to cope with speech mismatch.
c) to cope with device transfer size mismatch

* Caching

→ Fast memory holding copy of data.

→ Always just a copy. → Key to performance.

* Spooling

→ hold output for a device.

→ if device can serve only one request at a time.

→ For example : Printing.

* Device reservation

→ Provide exclusive access to a device.

→ system calls for allocation and deallocation.

→ watch out for ~~deadlock~~ deadlock.

Error handling

→ OS can recover from disk read, device unavailable, transient write failures.

→ most return an error number or code when I/o request fails.

→ system error logs hold problem reports.

Kernel data structures

→ Kernel keeps state info for I/o components, including open file tables, ; network connections.

* Consider reading a file from disk for a process?

→ Determine device holding file.
→ Translate name to device representation.
→ Physically read data from disk into buffer.
→ make data available to requesting process.
→ return control to process

streams

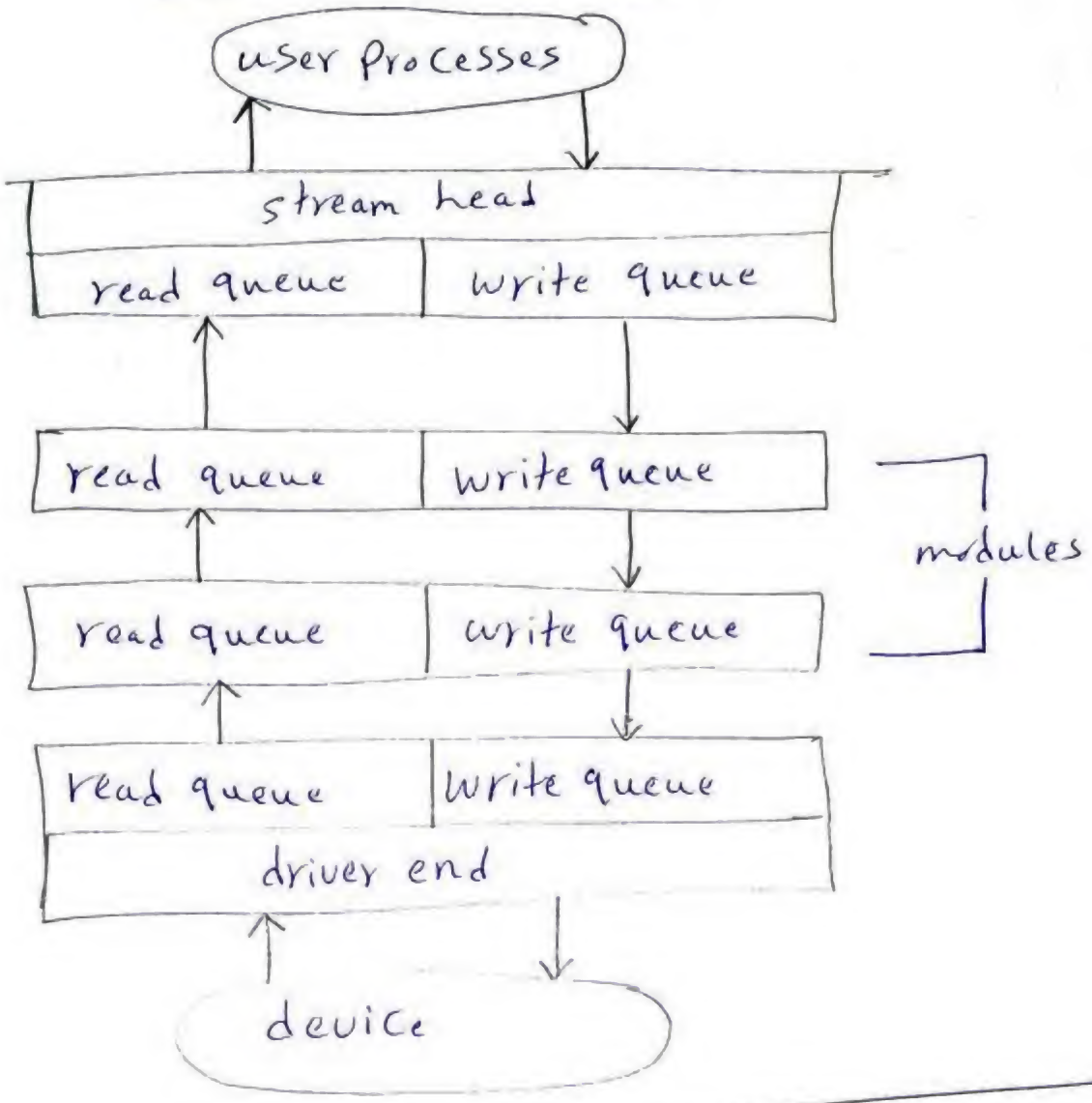
→ full-duplex communication channel between user-level process and a device.

→ Consists of:-
a) stream head interfaces with user process.
b) driver end interfaces with the device.
c) Zero or more stream modules between them.

→ Each module contains read queue & write queue.

⇒ message passing is used to communicate between queues.

Stream structure



Performance

Ilo major factor in system Performance:-

→ Demands CPU to execute device driver, Kernel Ilo code.

→ Context switches due to interrupts.

→ Data copying.

Improving Performance

1) Reduce data copying. 2) use DMA.
3) Reduce interrupt by using large transfers, smart controllers, Polling.
4) Balance CPU, memory, bus, and I/O performance ~~for highest throughput~~.

← توجد مجموعة رسومات عجيبة، ومدهشة المعقول ! ستبدأ بها جواسيلة العقل البشري.